International Journal of Computer Science Engineering
and Information Technology Research (IJCSEITR)
ISSN(P): 2249-6831; ISSN(E): 2249-7943
Vol. 4, Issue 1, Feb 2014, 73-82

COMPARATIVE EVALUATION FOR PERFORMANCE OF ANN TECHNIQUES IN BREAST CANCER DETECTION: AN INVESTIGATIVE APPROACH TO IMPROVEMENT

G ANJAN BABU & G SUMANA

SV University, Tirupathi, Andhra Pradesh, India

ABSTRACT

There is much research on medical diagnosis of breast cancer using WBCD data in neural network literature. In this paper the WBCD dataset is applied to the different networks for comparative evaluation of performance of ANN Techniques in Breast Cancer Detection as an investigative approach to improvement.

KEYWORDS: ANN Techniques, Breast Cancer Detection, WBCD

INTRODUCTION

Neural Networks are currently a 'hot' Research area in medicine, particularly in the fields of radiology, urology, cardiology, oncology and etc. It has a huge application in many areas such as education, business, medical, engineering and manufacturing. The main aim of research in medical diagnostics is to develop more cost-effective and easy–to-use systems, procedures and methods for supporting clinicians [1].

FEED FORWARD NEURAL NETWORK

A feed forward neural network is a biologically inspired classification algorithm. It consists of a (possibly large) number of simple neuron-like processing units, organized in layers. Every unit in a layer is connected with all the units in the previous layer. These connections are not all equal; each connection may have a different strength or weight. The weights on these connections encode the knowledge of a network. Often the units in a neural network are also called nodes. Data enters at the inputs and passes through the network, layer by layer, until it arrives at the outputs. During normal operation, that is when it acts as a classifier, there is no feedback between layers. Hence they are called feed forward neural networks.

Feed Forward Neural Network in Breast Cancer Detection

In Feed forward neural networks, the neurons are arranged in layers, with the first layer taking in inputs and the last layer producing outputs. The middle layers have no Connection with the external world, and hence are called hidden layers. Each neuron in one layer is connected to every neuron on the next layer. Hence information is constantly feed forward from one layer to the next. There is no connection among neurons in the same layer [2]. Learning in feed forward networks belongs to the realm of supervised learning, in which pairs of input and output values are fed into the network for many cycles, so that the network learns the relationship between the input and output. In Back propagation learning, every time an input training vector of a training sample is presented, the output vector o is compared to the desired value d. The comparison is done by calculating the squared difference of the two in equation (1):

$$Error = (d - o)^2 \quad (1)$$

The value of *Err* tells how far away from the desired value for a particular input. The goal of back propagation is to minimize the sum of Err for all the training samples, so that the network behaves in the most desirable way. We can

express the *Error* (*Err)* in terms of the input vector (i), the weight vector (w), and the threshold function of the neurons. Using a continuous function as the threshold function, the gradient of *Err* with respect to the w in terms of w and i can be expressed. Given the fact that decreasing the value of w in the direction of the gradient leads to the most rapid decrease in *Err*, the weight vectors updated every time a sample is presented.

Feed Forward Neural Network with Back Propagation Algorithm

The feed forward back propagation neural network can learn a function of mapping inputs to outputs by being trained with cases of input-output pairs. Back propagation neural network (BPNN) is actually a descending slope method to minimize the total square of the output, calculated by the network. There are three phases in the training process: first is to send the signal pattern forward, second is to calculate the propagated error and the last is to update all weights in the network. In addition BPNN also have the advantages of faster learning in multilayer Neural Network. The neurons in feed forward networks can be any transfer function of the designer wishes to use [3]. The network performance and convergence depends on many parameters like initial weights, learning rate and momentum used, number of nodes in the hidden layer during the training process.

FUZZY LOGIC

Zadeh introduced the theory of fuzzy logic in the late 1960s. Formerly Lukasiewicz had created the multi valued logic and the fuzzy logic is considered a rediscovery of that approach. Since various real world scenarios could not be represented by two values the fuzzy set approach was introduced [4]. Fuzzy sets, fuzzy membership functions, and fuzzy rules form the elemental components of the fuzzy logic decision making systems. A membership function forms an analogous part of a fuzzy set.

Fuzzy logic is a computational paradigm that provides a mathematical tool for representing and manipulating information in a way that resembles human communication and reasoning processes. A fuzzy variable also called a linguistic variable is characterized by its name tag, a set of fuzzy values also known as linguistic values or labels, and the membership functions of these labels; these latter assign a membership value, m label to a given real value uR, within some predefined range (known as the universe of discourse. While the traditional definitions of Boolean logic operations do not hold, new ones can be defined. Three basic operations, and, or, and not, are defined in fuzzy logic as follows:

μA and B(u)=μA(u)^μB(u)=min{μA(u), μB(u)}

Applying Evolution to Fuzzy Modeling

Depending on several criteria including the available a priori knowledge about the system, the size of the parameter set, and the availability and completeness of input, output data—artificial evolution can be applied in different stages of the fuzzy parameters search. Three of the four types of fuzzy parameters can be used to define targets for evolutionary fuzzy modeling: structural parameters, connective parameters, and operational parameters [5].

Knowledge Tuning (Operational Parameters)

The evolutionary algorithm is used to tune the knowledge contained in the fuzzy system by finding membership function values. An initial fuzzy system is defined by an expert. Then, the membership function values are encoded in a genome, and an evolutionary algorithm is used to find systems with high performance. Evolution often overcomes the local-minima problem present in gradient descent-based methods.

Behavior Learning (Connective Parameters)

In this approach, one supposes that extant knowledge is sufficient in order to define the membership functions; this determines, in fact, the maximum number of rules. As the membership functions are fixed and predefined, this approach lacks the flexibility to modify substantially the system behavior. Furthermore, as the number of variables and membership functions increases, the curse of dimensionality becomes more pronounced and the interpretability of the system decreases rapidly [6].

Structure Learning (Structural Parameters)

Evolution approach has to deal with the simultaneous design of rules, membership functions, and structural parameters. Some methods use a fixed-length genome encoding a fixed number of fuzzy rules along with the membership function values. Some structural constraints according to the available knowledge of the problem characteristics. Other methods use variable-length genomes to allow evolution to discover the optimal size of the rule base. In the WBCD example, evolutionary structure learning is carried out by encoding within the genome an entire fuzzy system Structure learning permits to specify other criteria related to the interpretability of the system, such as the number of membership functions and the number of rules.

Three Approaches to Behavior and Structure Learning

Both connective and structural parameters modeling can be viewed as rule base learning processes with different levels of complexity. They can thus be assimilated within other methods from machine learning, taking advantage of experience gained in this latter domain. In the evolutionary algorithm community there are two major approaches for evolving such rule systems: the Michigan approach and the Pittsburgh approach A more recent method has been proposed specifically for fuzzy modeling: the iterative rule learning approach. These three approaches are presented below [7].

The Michigan Approach

Each individual represents a single rule. The fuzzy inference system is represented by the entire population. Since several rules participate in the inference process, the rules are in constant competition for the best action to be proposed, and cooperate to form an efficient fuzzy system [8]. The cooperative–competitive nature of this approach renders difficult the decision of which rules are ultimately responsible for good system behavior. It necessitates an effective credit assignment policy to ascribe fitness values to individual rules.

The Pittsburgh Approach

Here, the evolutionary algorithm maintains a population of candidate fuzzy systems, each individual representing an entire fuzzy system. Selection and genetic operators are used to produce new generations of fuzzy systems [9]. Since evaluation is applied to the entire system, the credit assignment problem is eschewed. This approach allows including additional optimization criteria in the fitness function, thus affording the implementation of multi-objective optimization. The main shortcoming of this approach is its computational cost, since a population of full-fledged fuzzy systems has to be evaluated each generation.

The Iterative Rule Learning Approach

As in the Michigan approach, each individual encodes a single rule. An evolutionary algorithm is used to find a single rule, thus providing a partial solution. The evolutionary algorithm is used iteratively for the discovery of new rules, until an appropriate rule base is built. To prevent the process from finding redundant rules a penalization scheme is applied

each time a new rule is added. This approach combines the speed of the Michigan approach with the simplicity of fitness evaluation of the Pittsburgh approach [10]. The other incremental rule base construction methods, it can lead to a non-optimal partitioning of the antecedent space.

Fuzzy Logic in Breast Cancer Detection

With these three of top-performance systems, which serve to exemplify the solutions found by our evolutionary approach, The first system, delineated in consists of three rules, Taking into account all three criteria of performance classification rate, number of rules per system, and average number of variables per rule this system can be considered the top one over all 120 evolutionary runs. It obtains 98.7% correct classification rate over the benign cases, 97.07% correct classification rate over the malignant cases3, and an overall classification rate of 97.8%.

A thorough test of this three-rule system revealed that the second rule is never actually used; in the fuzzy literature this is known as a rule that never fires, i.e. is triggered by none of the input cases. Thus, it can be eliminated altogether from the rule base, resulting in a two-rule system. It obtains 97.3% correct classification rate over the benign cases, 97.49% correct classification rate over the malignant cases, and an overall classification rate of 97.36%. Finally, the best one-rule system found through our evolutionary approach. It obtains 97.07% correct classification rate over the benign cases, 97.07% correct classification rate over the malignant cases, and an overall classification rate of 97.07%.

SUPPORT VECTOR MACHINES

The original SVM algorithm was invented by Vladimir N. Vapnik and the current standard incarnation (soft margin) was proposed by Vapnik and Corinna Cortes in 1995 [11]. In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the input, making it a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

SVM Learning Method

Support Vector Machines (SVMs) are a set of related supervised learning methods used for classification and regression.

Linear Classification: When used for classification, the SVM algorithm creates a hyper plane that separates the data into two classes with the maximum-margin. Given training examples labeled either a maximum-margin hyper plane is identified which splits the "yes" from the "no" training examples, such that the distance between the hyper plane and the closest examples (the margin) is maximized [12].

Regression: A version of a SVM for regression was proposed in 1997 by Vapnik, Steven Golowich, and Alex Smola. This method is called support vector regression (SVR). The model produced by support vector classification only depends on a subset of the training data, because the cost function for building the model does not care about training points that lie beyond the margin. Analogously, the model produced by SVR only depends on a subset of the training data, because the cost function for building the model ignores any training data that is close (within a threshold ε) to the model

prediction [13].

SVM in Breast Cancer Detection

Support Vector Machine method was used on the set of 683 samples of actual data. Additional set of data of 117 samples is generated using Neural Network. The Accuracy or Efficiency of the detection of Breast Cancer by ANN is evaluated by using the Magnitude of Relative Error which is calculated using formula:

MRE = Abs ((AD - DD) /AD)

Where

AD is Actual detection, DD is desired detection

Pred (0.25) gives % of input that were predicted with an MRE is less than 0.25.

Measure of Average Efficiency is Calculated Using

Pred(p) = if (MRE < 0.25,1,0) Pred(p) = K/N

Where N total no of Historical Data and K is number of cases output with MRE less than or equal to P. Various groups of training and testing data were formed and mean square error was found out. The classification rate of SVM as classified as 98.6%. Actual Output (2 for Benignant and 4 for Malignant) verses Desired Output for training and testing samples.

GENETIC ALGORITHM

Genetic algorithms have been widely used in science as adaptive algorithms for solving practical problems such as optimization and machine learning Because GAS can be used to solve problems with different search spaces and parameters, GAS have also been widely used in biology and medical applications. For breast cancer diagnosis, Sahiner *et al*.Investigated a new approach that included a genetic algorithm for image feature selection, and a linear discriminant classifier or a back propagation neural network in the task of differentiating regions of interest POIS) on mammograms as either mass or normal tissue. They concluded that GAS provides versatility in the design of linear or nonlinear classifiers without a trade-off in the effectiveness of the selected features.

Genetic Algorithm for Breast Cancer Detection

Genetic algorithms can be used to determine the interconnecting weights of the ANN (i.e., to evolve weights in a fixed-structured, three-layer *ANN).* Similar to the genetic algorithm as described by David Montana and Lawrence Davis applied a genetic algorithm as follows:

Step-1 Let a chromosome be a "vector" of all the interconnecting weights of the ANN. Initialize a population of chromosomes (i.e., weight vectors) with each weight being between -1.O and + 1.O.

Step-2 Evaluate the fitness of each chromosome in the population. In this study, maximum fitness was equivalent to minimum overall error measure E in the training set. Then, apply "Roulette Wheel Parent Selection" to choose parent chromosomes for mating.

Step-3 Apply the crossover operation by taking two parent chromosomes (Parent 1 and 2) to produce two offspring chromosomes (Child 1 and 2). First, copy all the weights to the output unit of Child 1 from Parent 1; and Child 2 from Parent 2, respectively. Then, copy all the weights to the odd and even hidden units of Child 1

from those of Parent 1 and *2*, respectively. Alternatively, copy all the weights to the odd and even hidden units of Child 2 from those *of* Parent 2 and 1, respectively.

Step-4 Apply the mutation operation by randomly selecting a non-input unit and, for each incoming weight to the unit, add a random value between -1.O and + 1.O to the weight.

Step-5 Delete members of parent chromosomes to make room for offspring chromosomes. Evaluate the offspring chromosomes and insert them into the population.

Step-6 Increase generation by one. Repeat step 2 through *5* until a specific generation has been reached. GAS was applied to evolve the inter-connecting weights in the *ANN*, although GAS may only yield near-optimal solutions because of the large search space (multidimensional error surface).

With these algorithms, the GA was to provide the inter-connecting weights for the ANNs**.** Although the GA trained *ANN* was found to converge faster than the *ANN* in the training set. The GAS may have advantages over conventional other training techniques depending on the specific problem being addressed. The classification rate of the genetic algorithm is calculated as 98.8%. The GA can be adopted to minimize the overall error measure or to maximize the area under the ROC curve **as** required in clinical situations.

BACK PROPAGATION ALGORITHM

Back propagation is a common method of training artificial neural networks so as to minimize the objective function. Arthur E. Bryson and Yu-Chi Ho described it as a multi-stage dynamic system optimization method in 1969 [14]. It wasn't until 1974 and later, when applied in the context of neural networks and through the work of Paul Werbos, David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams, that it gained recognition, and it led to a “renaissance” in the field of artificial neural network research.

It is a supervised learning method, and is a generalization of the delta rule. It requires a dataset of the desired output for many inputs, making up the training set. It is most useful for feed-forward networks. The term is an abbreviation for "backward propagation of errors". Back propagation requires that the activation function used by the artificial neurons be differentiable. The back propagation learning algorithm can be divided into two phases: propagation and weight update:

Phase 1: Propagation

Each propagation involves the following steps:

- Forward propagation of a training pattern's input through the neural network in order to generate the propagation's output activations.
- Backward propagation of the propagation's output activations through the neural network using the training pattern's target in order to generate the deltas of all output and hidden neurons.

Phase 2: Weight Update

For each weight-synapse follow the following steps:

- Multiply its output delta and input activation to get the gradient of the weight.
- Bring the weight in the opposite direction of the gradient by subtracting a ratio of it from the weight.

This ratio influences the speed and quality of learning; it is called the *learning rate*. The sign of the gradient of a weight indicates where the error is increasing; this is why the weight must be updated in the opposite direction.

Modes of Learning

There are two modes of learning to choose from: One is on-line (incremental) learning and the other is batch learning. In on-line (incremental) learning, each propagation is followed immediately by a weight update. In batch learning, much propagation occur before weight updating occurs. Batch learning requires more memory capacity, but on-line learning requires more updates.

Algorithm: Actual algorithm for a 3-layer network (only one hidden layer): Initialize the weights in the network (often randomly).

Do

For each example in the training set.

O = neural-net-output (network, e); forward pass

T = teacher output for e.

Calculate error (T - O) at the output units.

Compute delta_wh for all weights from hidden layer to output layer ; backward pass.

Compute delta_wi for all weights from input layer to hidden layer; backward pass. Continued

Update the weights in the network.

Until all examples classified correctly or stopping criterion satisfied. Return the network.

As the algorithm's name implies, the errors propagate backwards from the output nodes to the inner nodes. Technically speaking, back propagation calculates the gradient of the error of the network regarding the network's modifiable weights. This gradient is almost always used in a simple stochastic gradient descent algorithm to find weights that minimize the error. Often the term "back propagation" is used in a more general sense, to refer to the entire procedure encompassing both the calculation of the gradient and its use in stochastic gradient descent[15]. Back propagation usually allows quick convergence on satisfactory local minima for error in the kind of networks to which it is suited.

Back propagation networks are necessarily multilayer perceptrons (usually with one input, one hidden, and one output layer). In order for the hidden layer to serve any useful function, multilayer networks must have non-linear activation functions for the multiple layers: a multilayer network using only linear activation functions is equivalent to some single layer, linear network. Non-linear activation functions that are commonly used include the logistic function, the softmax function, and the Gaussian functions.

The back propagation algorithm for calculating a gradient has been rediscovered a number of times, and is a special case of a more general technique called automatic differentiation in the reverse accumulation mode.

Using this algorithm, the weight vectors are modified so that the value of Err for a particular input sample decreases a little bit every time the sample is presented. When all the samples are presented in turns for many cycles, the sum of Err gradually decreases to a minimum value, which is the goal of back propagation algorithm. Here Back propagation algorithm is used to train the network. With this good diagnostic performance is resulted.

DATASET

Breast cancer database is applied to the neural network techniques, which was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. The database contains 699 samples with 683 complete data and 16 samples with missing attributes. There are 9 integer-valued attributes and each data values range from 1 to 10, as follows [38]

- Lump Thickness;
- Uniformity of Cell Size;
- Uniformity of Cell Shape;
- Marginal Adhesion – fibrous bands tissue that form between two surfaces;
- Single Epithelial Cell Size – the size of a single cell that forms tissues that lines the outside of the body and the passageways that lead to or from the surface;
- Bare Nuclei;
- Bland Chromatin–evaluates for the presence of Barr bodies;
- Normal Nucleoli;
- Mitoses – cell growth.

These attributes measure the external appearance and internal chromosome changes in nine different scales. There are two values in the class variable of breast cancer: *benign (non-cancerous) and malignant (cancerous).*With the help of this database, applying to the different neural network techniques the results is achieved an shown in Table I.

Table 1: Results of ANN Techniques Applied to the WBCD Data Set

Sl. No.	ANN Technique	Accuracy	Sensitivity	Specificity
1.	Feed Forward Network	99.37%	96.3%	98.87%
2.	Fuzzy Logic	97.8%	96.3%	94.68%
3.	Support Vector Machine	95.39%	96.89%	94.2%
4.	Genetic Algorithm	98.8%	97.8%	96.7%
5.	Back Propagation	97.89%	98.67%	96.78%

CONCLUSIONS

Here the ANN techniques is applied to the WBCD dataset and calculated the accuracy, sensitivity, specificity among all these feed forward neural network is obtained best accuracy, when compared to other networks. Hence with these we conclude that feed forward neural network give the good performance for detecting the breast cancer with back propagation algorithm.

REFERENCES

1. Poulo J. Lisboa, Azzam F.G.Taktak ***"The use of artificial neural networks in decision support in cancer"*** published in Elsevier by, 2005.
2. S. Saheb Basha, DR.K.Satya Prasad, article ***"Automatic Detection of Breast Cancer Mass In Mammograms Using Morphological Operators And Fuzzy C –Means Clustering"*** (JATIT) Journal of Theoretical and Applied

Information Technology, published an by 2005.

3. Yuehui Chen, Ajith Abraham ***"Hybrid Neuro computing for Breast Cancer Detection",*** 2005.
4. Jernyhui.re.et.al., ***"Intelligent System for Breast Cancer Prognosis using Multiwavelet Packets and Neural Network",*** published in a Journal World Academy of Science, Engineering and Technology, 2005.
5. Sudhir D. Sawarkar, Ashok A.Ghatol, Amol P. Pande ***"Neural Network Aided Breast Cancer Detection and Diagnosis Using Support Vector Machine".*** 7th WSEAS International Conference on Neural Networks, Cavtat, Croatia, 2006, published a paper on
6. Marisa Dolled-Filhart, LISA Ryden, Melissa Cregger. ***"Classification of Breast Cancer Using Genetic Algorithms and Tissue Microarrays",*** in a journal AACR, 2006,
7. Yuehui Chen, Yan Wang, Bo Yang ***"Evolving Hierarchical RBF Neural Networks for Breast Cancer Detection"*** at Springer publications, 2006.
8. Gustavo Ferrero, Paola Britos, Ramon Garcia Martinez. ***"Detection of Breast Lesions in Medical Digital Imaging Using Neural Networks",*** 2006,
9. Kruger WM, Prof.Appfelstaedt JP. ***"Young breast cancer patients in the developing world: incidence, choice of surgical treatment and genetic factors"*** published in a journal CPD, 2007,
10. Afzan Adam, Khairuddin Omar ***"Computerized Breast Cancer Diagnosis with Genetic Algorithms and Neural Network",*** 2007.
11. G Heona Ioana Andreea, Ralucapegza, Luana lascu, Simona Bondari, Zoia Stoica, A. Bondari. ***"The Role of Imaging Techniques in Diagnosis of Breast Cancer"*** International Journal of Computer Science and Engineering (IJCSE), 2009.
12. Ajay Nagesh Baravanhalty, Shridar Gancean, Shannan Agner, James Peter Monaco, ***"Coputerised image based detection and grading of lymphocytic infiltration in HER2 breast cancer histopathology"***, IEEE Transactions on biomedical engineering vol.57, No-3, March -2010.
13. Jean Roman Dalle, Wee Kheng Leow, Daniel Recoeanis, Adima Eunice Tatac, Thomas C. Putti, ***"Automatic breast cancer grading of histopathological images"***, 30th annual international IEEE EMBS conference van couver British Columbia, Canada, august, 2008.
14. Ali H. Al-Timemy, Fawzi M. Al-Naima, Nebras H. Qaeeb, ***"Probabilistic Neural Network for Breast Biopsy Classification"***, MASAUM Journal of Computing, Volume 1 Issue 2, September 2009.
15. Hoque A, Lippman Scott M, Boiko Iouri V, Atkinson Edward N, Nour Sneige, Aysegul Sahin, Weber Diane M, Seymon Risin, Lagios Michael D, Roland Schwarting, Colburn William J, Kapil Dhingra, Michele Follen, Kelloff Gary J, Boone Charles W, Hittelman Walter N: Quantitative ***"Nuclear Morphometry by Image Analysis for Prediction of Recurrence of Ductal Carcinoma in Situ of the Breast. Cancer Epidemiol Biomarker"*** Prev 2001, 10:249-259.